9A

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/602,201	06/24/2003	Marc Shapiro	MS1-1509US	2205

22971 7590 01/12/2006

MICROSOFT CORPORATION
ATTN: PATENT GROUP DOCKETING DEPARTMENT
ONE MICROSOFT WAY
REDMOND, WA 98052-6399

EXAMINER
MORRISON, JAY A

ART UNIT	PAPER NUMBER
2168	

DATE MAILED: 01/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/602,201	Applicant(s) SHAPIRO ET AL.	
	Examiner Jay A. Morrison	Art Unit 2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 June 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-40* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 June 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *3/18/2005*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-40 are pending

Specification

2. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Objections

3. Claims 25-30 are objected to because of the following informalities: "processor-readable medium" taught in claims is not disclosed in specification and therefore does not provide antecedent basis for claim terminology. Appropriate correction is required.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-8, 11-18, 22-23, 25-38 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The cited claims do not produce a useful, concrete and tangible result.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-40 are rejected under 35 U.S.C. 102(b) as being anticipated by Preguica et al ("Efficient semantics-aware reconciliation for optimistic write sharing", Microsoft Corporation, May 2002).

With respect to claim 1, Preguica teaches

"receiving a list of candidate primitive actions comprising primitive actions related to a change in a file system state of a first file system, and primitive actions related to a change in a file system state of a second file system, both file systems constituting replicas of a shared file system" (pages 3-4, section 2.1, whereas Preguica's logs are equivalent to the claimed list of actions related to changes in the state of a file system);

"and generating a schedule of non-conflicting primitive actions comprising one or more primitive actions from the list" (pages 3-4, section 2.1, whereas Preguica's IceCube proposing schedules combining actions is equivalent to the claimed generation of a schedule of actions).

With respect to claim 2, Preguica teaches

"the generating a schedule comprises: selecting a candidate primitive action from the list based on a value assessment of the candidate primitive actions" (page 6, section 3, whereas Preguica's IceCube system selecting the highest-valued ones is equivalent to the claimed selecting action from list based on value assessment).

With respect to claim 3, Preguica teaches

"the generating a schedule comprises: selecting a candidate primitive action from the list based on a value assessment of the candidate primitive actions" (page 10, section 4.1, whereas Preguica's selecting candidates based on merit estimator value is equivalent to the claimed selecting action from list based on value assessment);

"and executing the selected candidate primitive action starting from a checkpoint file system state" (page 12, section 4.1, whereas Preguica's executing against a given data state is equivalent to the claimed executing the action starting from checkpoint state);

"and scheduling the selected candidate primitive action if the executing the selected candidate primitive is successful" (page 12, section 4.1, whereas Preguica appending to the schedule if successful is equivalent to the claimed scheduling the candidate if successful).

With respect to claim 4, Preguica teaches

"receiving a log constraint representing a relationship between two of the primitive actions of the first file system, or between two of the primitive actions of the second file system" (pages 7-8, section 3.3.1).

With respect to claim 5, Preguica teaches

"the log constraint comprises a parcel relationship, a predecessor-successor relationship, or an alternatives relationship" (pages 7-8, section 3.3.1).

With respect to claim 6, Preguica teaches

"receiving an object constraint representing a relationship between one of the primitive actions of the first file system and one of the primitive actions of the second file system" (page 8, section 3.3.2).

With respect to claim 7, Preguica teaches

"the object constraint comprises a mutually-exclusive relationship or a best-order relationship" (page 8, section 3.3.2).

With respect to claim 8, Preguica teaches

"the object constraint is based on application semantics related to the primitive actions of the shared file system" (page 8, section 3.3.2).

With respect to claim 9, Preguica teaches

"proposing the generated schedule to a user" (pages 4-5, section 2.2, whereas Preguica's scheduler proposing is equivalent to the claimed proposing the generated scheduler).

With respect to claim 10, Preguica teaches

"committing the generated schedule on at least one of the first file system and the second file system" (pages 4-5, section 2.2, whereas user accepting to commit the schedule is equivalent to the claimed committing the generated schedule).

With respect to claim 11, Preguica teaches

"selecting one of the scheduled primitive actions to undo" (pages 4-5, section 2.2, whereas Preguica's using the graphical user interface to adding or removing constraints is equivalent to the claimed selecting of actions to undo);

"and undoing the selected scheduled primitive action" (pages 4-5, section 2.2, whereas Preguica's adding or removing constraints is equivalent to the claimed undoing of selected actions).

With respect to claim 12, Preguica teaches

"selecting one of the scheduled primitive actions to undo" (pages 4-5, section 2.2, whereas Preguica's using the graphical user interface for adding or removing constraints is equivalent to the claimed selecting of actions to undo);

"undoing the selected scheduled primitive action" (pages 4-5, section 2.2, whereas Preguica's adding or removing constraints is equivalent to the claimed undoing of selected actions);

"and undoing another of the primitive actions that depends on the selected scheduled primitive action to undo" (pages 4-5, section 2.2, whereas Preguica's parcel of actions are equivalent to the claimed actions depending on the selected action).

With respect to claim 13, Preguica teaches

"the receiving the list comprises: decomposing a file system command at the first file system into the primitive actions related to a change in a file system state" (pages 7-8, section 3.3.1, whereas Preguica's decomposing a complex action into more primitive ones is equivalent to the claimed decomposing file system commands into primitive actions).

With respect to claim 14, Preguica teaches

"the generating comprises: selecting a candidate primitive action from the list" (page 10, section 4.1, whereas Preguica's selecting candidates is equivalent to the claimed selecting action from list);

"executing the selected candidate primitive action starting from a checkpoint file system state" (page 12, section 4.1, whereas Preguica's executing against a given data state is equivalent to the claimed executing the action starting from checkpoint state);

"determining whether the executing the selected candidate primitive action was successful; and if the executing the selected candidate primitive action was successful, adding the selected candidate primitive action to the schedule" (page 12, section 4.1, whereas Preguica appending to the schedule if successful is equivalent to the claimed scheduling the candidate if successful).

With respect to claim 15, Preguica teaches

"the generating further comprises: if the executing the selected candidate primitive action was successful, removing the selected candidate primitive action from the list, removing from the list another candidate primitive action that may execute only before the selected candidate primitive action, and removing from the list another candidate primitive action that conflicts with the selected candidate primitive action" (page 13, section 4.2.2, whereas Preguica's if post-condition fails the action and implying set being removed is equivalent to the claimed if executing action not successful then removing actions).

With respect to claim 16, Preguica teaches

"the generating further comprises: if the executing the selected candidate primitive action was not successful, removing the selected candidate action from the list, and removing from the list all actions that are in a parcel with the selected candidate action" (page 13, section 4.2.2, whereas Preguica's if post-condition fails the action and implying set being removed is equivalent to the claimed if executing action not successful then removing actions).

With respect to claim 17, Preguica teaches

"the generating further comprises: if the executing the selected candidate primitive action was not successful, rolling back side effects that resulted from executing the selected candidate action, and rolling back side effects that resulted from previously executing other actions that are in a parcel with the selected candidate action" (page 13, section 4.2.2, whereas Preguica's using record states and roll back is equivalent to the claimed rolling back side effects).

With respect to claim 18, Preguica teaches

"receiving a first file system command; receiving a second file system command" (pages 4-5, section 2.2, and figure 2, whereas Preguica's flight reservations are equivalent to the claimed first and second file system commands);

"decomposing the first file system command into one or more corresponding first primitive actions; decomposing the second file system command into one or more corresponding second primitive actions" (Page 7-8, section 3.3.1, whereas Preguica's decomposing a complex action into primitive ones is equivalent to the claimed decomposing of file system commands);

"receiving an object constraint indicating a relationship between a selected one of the first primitive actions and a selected one of the second primitive actions" (pages 4-5, section 2.2, and figure 2, whereas Preguica's constraints on flight reservation actions are equivalent to the claimed first and second file system commands);

"and if the object constraint indicates mutual exclusion, scheduling either the selected one of the first primitive actions or the selected one of the second primitive actions in a schedule of non-conflicting primitive actions based on the object constraint otherwise, scheduling both the selected one of the first primitive actions and the selected one of the second primitive actions" (page 8, section 3.3.2, and table 7, whereas Preguica's mutuallyExclusive object constraint is equivalent to the claimed object constraint).

With respect to claim 19, Preguica teaches

"executing the selected one of the first primitive actions and the selected one of the second primitive actions on a file system with the object constraint to determine whether the selected first primitive action and the selected second primitive action conflict" (page 12, section 4.2.1, whereas Preguica's scheduleOne procedure creating and executing a single schedule and testing schedules success is equivalent to the claimed executing the selected actions to determine if they conflict).

With respect to claim 20, Preguica teaches

"logging the first primitive actions in a first action log" (pages 4-5, section 2.2, whereas Preguica's A's log is equivalent to the claimed logging of actions in first log);

"logging the second primitive actions in a second action log" (pages 4-5, section 2.2, whereas Preguica's B's log is equivalent to the claimed logging of actions in second log);

"combining the first action log and the second action log into a reconciliation log" (pages 4-5, section 2.2, and figure 2, whereas Preguica's reconciler window is equivalent to the claimed combination of first and second action logs);

"selecting a primitive action from the reconciliation log" (pages 4-5, section 2.2, whereas Preguica's user selecting certain actions is equivalent to the claimed selecting of actions from the reconciliation log);

"and executing the selected primitive action on a file system" (pages 4-5, section 2.2, whereas Preguica's user committing the schedule to be replayed is equivalent to the claimed executing of selected action).

With respect to claim 21, Preguica teaches

"the selecting a primitive action comprises: selecting a primitive action from the reconciliation log that has a higher value that the other primitive actions in the reconciliation log" (pages 9-10, section 4.1, whereas Preguica's scheduler selecting candidates with highest merit is equivalent to the claimed selecting action that has a higher value).

With respect to claim 22, Preguica teaches

"the relationship indicated by the object constraint comprises a mutually exclusive relationship" (page 8, section 3.3.2).

With respect to claim 23, Preguica teaches

"identifying a primitive action that conflicts with the scheduled primitive action among the first primitive actions and the second primitive actions; and excluding the identified conflicting primitive action from the schedule of non-conflicting primitive actions" (page 13, section 4.2.2, whereas Preguica's if post-condition fails the action and implying set being removed is equivalent to the claimed identifying and excluding conflicting action from the schedule).

With respect to claim 24, Preguica teaches

"the selecting a primitive action comprises: attributing a higher value to a first primitive action than a second primitive action if scheduling the first primitive action would result in fewer conflicts with other primitive actions in the reconciliation log than would scheduling the second primitive action" (page 13, section 4.1, whereas Preguica's when an action fails dynamically its merit dropping drastically is equivalent to the claimed attributing of a higher value to the first action than a second if scheduling results in fewer conflicts).

With respect to claim 25, Preguica teaches

"receiving a first file system command to change the state of a first file system" (pages 4-5, section 2.2, whereas Preguica's A's tentative requests are equivalent to the claimed receiving of a command to change the state of the first system);

"generating a plurality of first primitive actions corresponding to the first file system command" (pages 7-8, section 3.3.1, whereas Preguica's decomposing a complex action into more primitive ones is equivalent to the claimed generation of a plurality of primitive actions corresponding to the command);

"and receiving one or more log constraints representing a relationship between two of the plurality of first primitive actions" (pages 4-5, section 2.2, whereas Preguica's A's constraints are equivalent to the claimed received log constraints);

"and scheduling one or more of the first primitive actions in a non-conflicting schedule of primitive actions based on the one or more log constraints" (pages 6-7, section 3.3, and table 1, whereas Preguica's constraint scheduling relations are equivalent to the claimed scheduling of actions in a non-conflicting schedule).

With respect to claim 26, Preguica teaches

"receiving a plurality of second primitive actions corresponding to a second file system command to change the state of a second file system" (page 4-5, section 2.2, whereas Preguica's B's logs are equivalent to the claimed plurality of second actions);

"selecting one of the first primitive actions and one of the second primitive actions" (pages 6-7, section 3.3, and table 1, whereas Preguica's actions a and b are equivalent to the claimed first and second actions);

"receiving an object constraint representing a relationship between the selected first primitive action and the selected second primitive action" (pages 6-7, section 3.3, and table 1, whereas Preguica's static constraints are equivalent to the claimed object constraints);

"and scheduling the selected first primitive action or the selected second primitive action based on the object constraint" (pages 6-7, section 3.3, and table 1, whereas Preguica's relations schedules are equivalent to the claimed scheduling of actions based on object constraints).

With respect to claim 27, Preguica teaches

"the one or more log constraints comprise user constraints" (pages 7-8, section 3.3.1).

With respect to claim 28, Preguica teaches

"the one or more log constraints comprise application constraints" (pages 7-8, section 3.3.1).

With respect to claim 29, Preguica teaches

"committing the schedule of non-conflicting actions on the first file system" (pages 4-5, section 2.2, whereas Preguica's user choosing to commit the schedule is equivalent to the claimed committing the schedule).

With respect to claim 30, Preguica teaches

"the object constraint depends upon application semantics" (page 8, section 3.3.2).

With respect to claim 31, Preguica teaches

"an input/output module receiving a file system command causing a tentative update to a file system state" (page 18-19, section 5.4);

"a reconcilable file system operable to receive the file system command, and generate a plurality of primitive actions representing the file system command and a log constraint representing a relationship between two primitive actions in the plurality of primitive actions; and a log receiving the plurality of primitive actions and the log constraint" (page 18-19, section 5.4).

With respect to claim 32, Preguica teaches

"a decomposition module operable to decompose the file system command into the plurality of primitive actions" (Page 7-8, section 3.3.1, whereas Preguica's decomposing a complex action into primitive ones is equivalent to the claimed decomposing of file system commands);

"and a recording module operable to receive the plurality of primitive actions and record the plurality of primitive actions in the log" (pages 4-5, section 2.2, whereas Preguica's A's log of tentative actions are equivalent to the claimed plurality of actions recorded in the log).

With respect to claim 33, Preguica teaches

"the decomposition module is further operable to generate the log constraint and conmunicate the log constraint to the recording module" (pages 4-5, section 2.2, whereas Preguica's A adding constraints is equivalent to the claimed generating log constraints).

With respect to claim 34, Preguica teaches

"the log constraint is a user constraint" (pages 7-8, section 3.3.1).

With respect to claim 35, Preguica teaches

"the log constraint is an application constraint" (pages 7-8, section 3.3.1).

With respect to claim 36, Preguica teaches

"the log constraint is a parcel constraint indicating that the two primitive actions must be executed together" (pages 7-8, section 3.3.1).

With respect to claim 37, Preguica teaches

"the log constraint is a predecessor-successor constraint indicating that the two actions must be executed in a prescribed order" (pages 7-8, section 3.3.1).

With respect to claim 38, Preguica teaches

"the log constraint is an alternatives constraint indicating that only one of the two actions must be selected and executed" (pages 7-8, section 3.3.1).

With respect to claim 39, Preguica teaches

"the reconcilable file system is further operable to receive a schedule of non-conflicting primitive actions and commit the non-conflicting primitive actions to the file system" (pages 4-5, section 2.2, whereas Preguica's top pane including proposed schedule and committing the schedule is equivalent to the claimed receiving a schedule and committing the actions).

With respect to claim 40, Preguica teaches

"the reconcilable file system is further operable to roll back changes that resulted in the tentative file system state in order to commit the schedule of non-conflicting primitive" (page 13, section 4.2.2, whereas Preguica's using record states and roll back is equivalent to the claimed rolling back side effects).

Conclusion

6. The prior art made of record, listed on form PTO-892, and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jay Morrison whose telephone number is (571) 272-7112. The examiner can normally be reached on Monday to Thursday from 7:30am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin, can be reached at (571) 272-4146 or TC 2100 customer service can be contacted at (703) 306-5631. The fax phone numbers for the organization where this application or proceeding is assigned are (571) 273-8300 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9600.

Jay Morrison
Assistant Examiner
TC2100

Jeffrey Gaffin
Primary Examiner
TC2100